University of Notre Dame
Aerospace and Mechanical Engineering

AME 469: Introduction to Robotics
Homework 5 Solutions

B. Goodwine
Spring, 2001

1. (Craig, 3.11)
   Following the usual rules for affixing frames to manipulators results in the frames illustrated in Figure 1.

   Consulting the figure, the following link parameters are apparent:

i	α_{i-1}	a_{i-1}	d_i	θ_i
4	0	0	0	θ_4
5	ϕ	0	0	θ_5
6	$-\phi$	0	0	θ_6

   Determining ${}^B_4T$,${}^4_5T$ and ${}^5_6T$ is now a simple matter of substituting these values into Equation 3.6, or, even simpler, using the Mathematica function from the previous homework.

```
<</home/bill/courses/me469/math/forward.m
T[0,0,0,t4] . T[phi,0,0,t5] . T[-phi,0,0,t6] //Simplify
```

   produces the mess

```
{-(Sin[t4]*(Cos[phi]*Cos[t6]*Sin[t5] +
         Cos[phi]^2*Cos[t5]*Sin[t6] + Sin[phi]^2*Sin[t6])) +
    Cos[t4]*(Cos[t5]*Cos[t6] - Cos[phi]*Sin[t5]*Sin[t6]),
   -(Cos[phi]^2*Cos[t5]*Cos[t6]*Sin[t4]) -
    Cos[t6]*Sin[phi]^2*Sin[t4] -
    Cos[phi]*Cos[t4 + t6]*Sin[t5] - Cos[t4]*Cos[t5]*Sin[t6],
   -(Sin[phi]*(Cos[phi]*(-1 + Cos[t5])*Sin[t4] +
        Cos[t4]*Sin[t5])), 0},
  {Cos[phi]*Cos[t4 + t6]*Sin[t5] +
    Cos[t4]*Sin[phi]^2*Sin[t6] +
    Cos[t5]*(Cos[t6]*Sin[t4] + Cos[phi]^2*Cos[t4]*Sin[t6]),
   Cos[phi]^2*Cos[t4]*Cos[t5]*Cos[t6] +
    Cos[t4]*Cos[t6]*Sin[phi]^2 - Cos[t5]*Sin[t4]*Sin[t6] -
    Cos[phi]*Sin[t5]*Sin[t4 + t6],
```

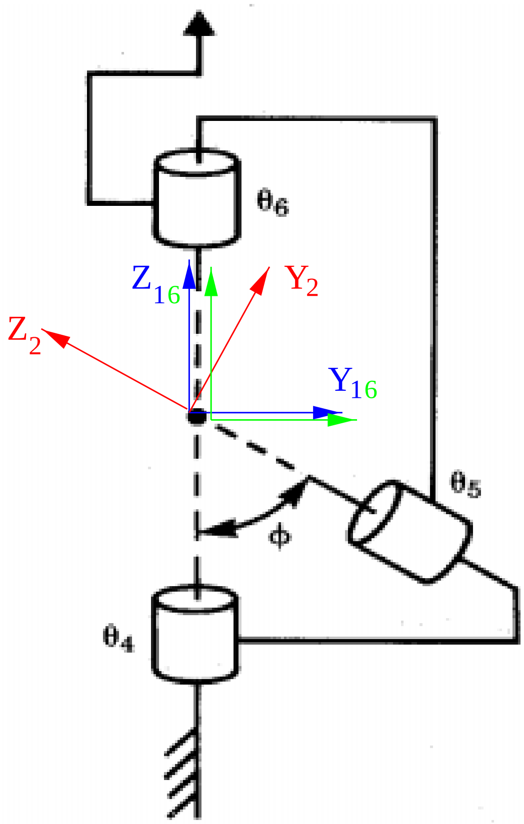

Figure 1. Frames for problem 1.

```
 Sin[phi]*(Cos[phi]*Cos[t4]*(-1 + Cos[t5]) -
    Sin[t4]*Sin[t5]), 0},
{Sin[phi]*(Cos[t6]*Sin[t5] +
    Cos[phi]*(-1 + Cos[t5])*Sin[t6]),
 Sin[phi]*(Cos[phi]*(-1 + Cos[t5])*Cos[t6] -
    Sin[t5]*Sin[t6]), Cos[phi]^2 + Cos[t5]*Sin[phi]^2, 0},
{0, 0, 0, 1}}
```

To check to make sure it makes sense, take all the joint angles, θ_4, θ_5 and θ_6 to be zero:

```
<</home/bill/courses/me469/math/forward.m
T[0,0,0,0] . T[phi,0,0,0] . T[-phi,0,0,0] //Simplify //MatrixForm
```

which gives the expected answer

$$ {}^B_6T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

If $\theta_4 = \pi$ and the other joints are zero, then

```
<</home/bill/courses/me469/math/forward.m
T[0,0,0,Pi] . T[phi,0,0,0] . T[-phi,0,0,0] //Simplify //MatrixForm
```

$$ {}^B_6T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

which also is expected because rotating joint 4 by π will rotate the x and y components of a vector by 180°, which will make their components negative of what they start as.

2. (Craig, 3.16)

The link frame assignments are illustrated in Figure 2 (any axis not shown is determined by the right hand rule).

Consulting the figure, and naming the distances a_1 and a_2, the following link parameters are apparent:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	$\frac{\pi}{2}$	a_1	d_2	0
3	$-\frac{\pi}{2}$	0	0	θ_3

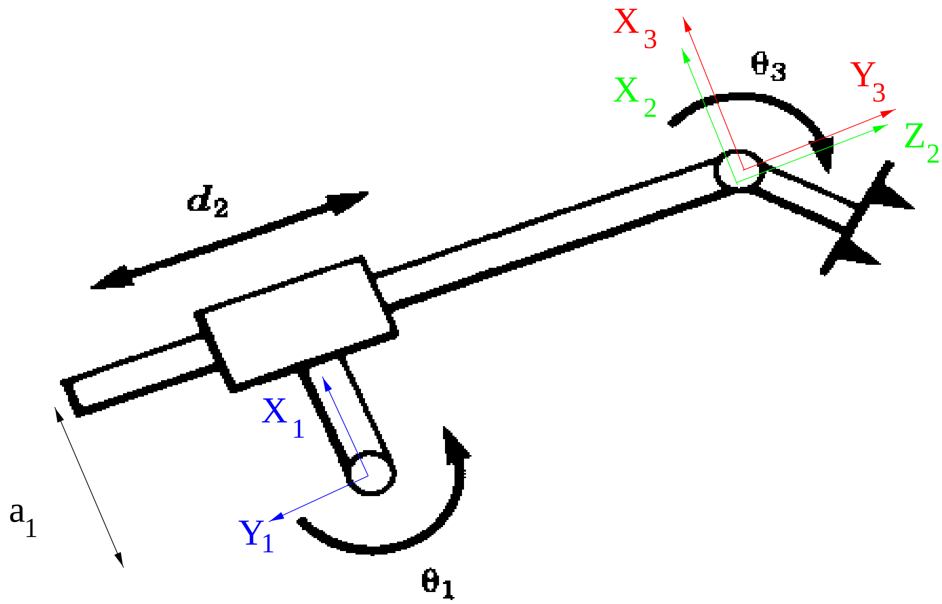

Figure 2. Link frame assignments for problem 2.

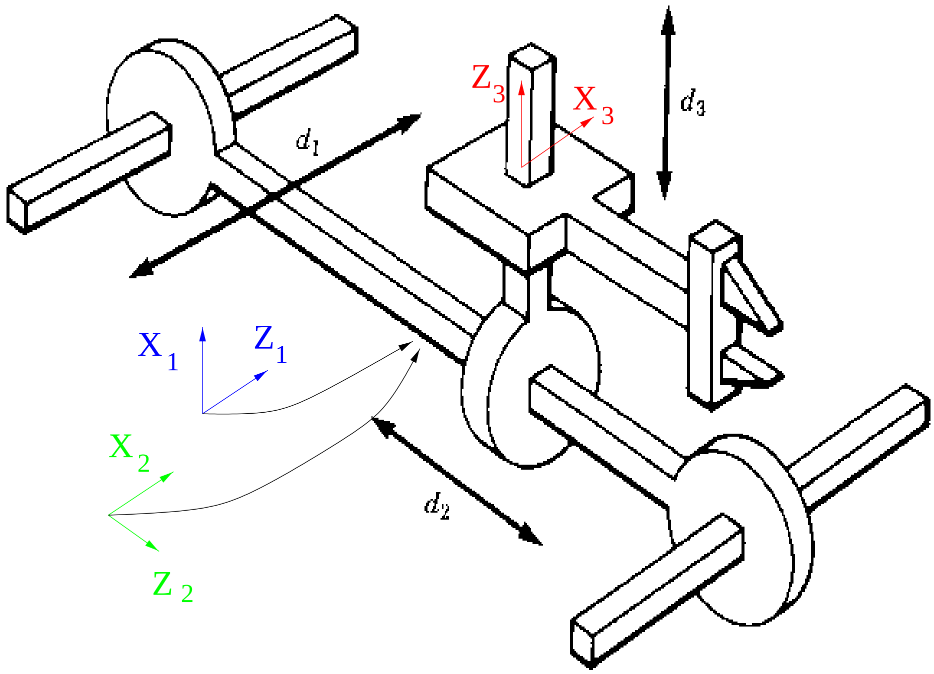

Figure 3. Link frame assignments for problem 3.

Note that I took the the reference value for θ_3 to be the position of the wrist when it is "straight out," and the reference value for d_2 to be zero when the frame 3 is located a distance of a_2 away from frame 2.

Plugging into Equation 3.6, or the Mathematica function gives the forward kinematics

$$ {}^0_3T = \begin{bmatrix} \cos(\theta_1+\theta_3) & -\sin(\theta_1+\theta_3) & 0 & a_1\cos\theta_1 + d_2\sin\theta_1 \\ \sin(\theta_1+\theta_3) & \cos(\theta_1+\theta_3) & 0 & -d_2\cos\theta_1 + a_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

3. (Craig, 3.21)

Note that I picked joint axis 1 in the middle of the mechanism. You could have picked the joint axis to be any line parallel to the two rails that define axis 1.

The link frame assignments are illustrated in Figure 3.

Consulting the figure, the following link parameters are apparent:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	0
2	$-\frac{\pi}{2}$	0	d_2	$-\frac{\pi}{2}$
3	$-\frac{\pi}{2}$	0	d_3	0

Plugging into Equation 3.6, or the Mathematica function gives the forward kinematics

$$
{}^0_3T = \begin{bmatrix} 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

4. This problem was to compute the inverse kinematics of the manipulator illustrated in Craig, Figure 3.29. To compute the *forward* kinematics, you could have either assigned frames and then computed the location of the tip of the manipulator in the zero frame (like problem 3.9), or, in this case the manipulator is simple enough that it is possible to simply compute the tip location based on geometry.

   Either method gives

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta_1 \left(L_1 + L_2\cos\theta_2 + L_3\cos(\theta_2+\theta_3)\right) \\ \sin\theta_1 \left(L_1 + L_2\cos\theta_2 + L_3\cos(\theta_2+\theta_3)\right) \\ L_2\sin\theta_2 + L_3\sin(\theta_2+\theta_3) \end{bmatrix}.
$$

   This assumes that the reference frame is as illustrated in Figure 4.

   If we let $L_1 = 1$, $L_2 = 1$ and $L_3 = 0.5$, and testing it on a solution that is obvious (where $(x, y, z) = (2, 0, 0.5)$), using the Mathematica FindRoot[] function gives

```
l1 = 1;
l2 = 1;
l3 = 0.5;
x = 2;
y = 0;
z = 0.5;
FindRoot[{x == Cos[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    y == Sin[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    z == l2 Sin[t2] + l3 Sin[t2 + t3]}, {t1, 0.1},{t2, 0.1},{t3, 1.5}]
```

   gives $\{t_1 \to 1.18241\,10^{-11}, t_2 \to -1.114\,10^{-11}, t_3 \to 1.5708\}$ which is clearly correct.

   Checking with another desired point, $(x, y, z) = (0.8, -0.776, 1.1)$ which returns

```
l1 = 1;
l2 = 1;
l3 = 0.5;
```

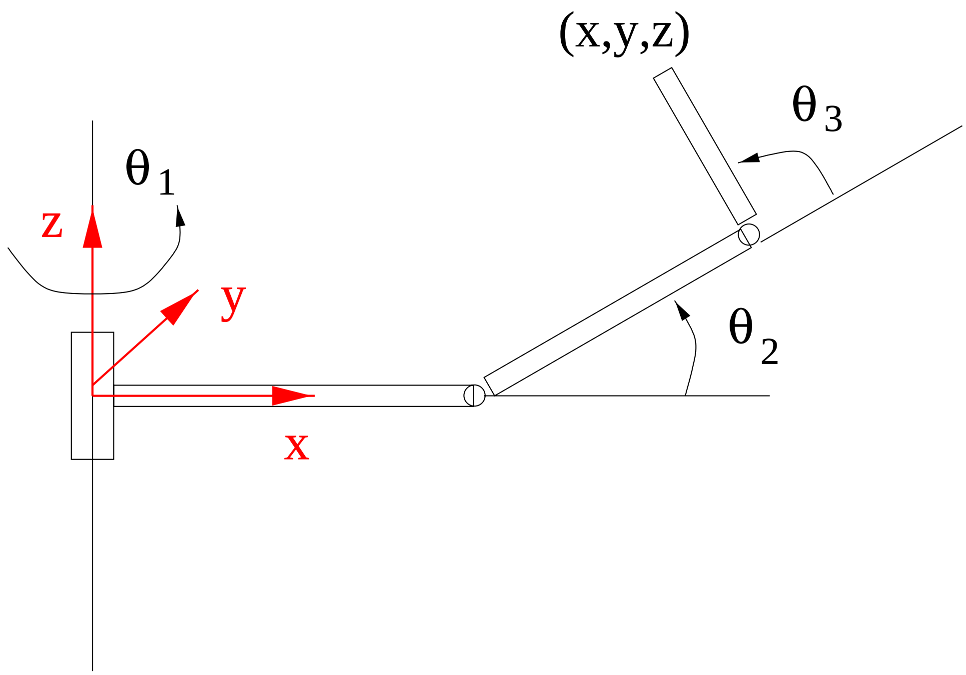

Figure 4. Figure for Problem 4.

```
x = 0.8;
y = -0.776;
z = 1.1;
FindRoot[{x == Cos[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    y == Sin[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    z == l2 Sin[t2] + l3 Sin[t2 + t3]}, {t1, 0.1},{t2, 0.1},{t3, 1.5}]
```

which returns $\{t_1 \to -0.770171, t_2 \to 0.998116, t_3 \to 1.59768\}$.

Finally, if we try to specify a point that is not reachable, Mathematica complains:

```
l1 = 1;
l2 = 1;
l3 = 0.5;
x = 1.5;
y = -1.5;
z = 1.1;
FindRoot[{x == Cos[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    y == Sin[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    z == l2 Sin[t2] + l3 Sin[t2 + t3]}, {t1, 0.1},{t2, 0.1},{t3, 1.5}]

FindRoot::"cvnwt":
    "Newton's method failed to converge to the prescribed accuracy after
15 iterations."
```

Even increasing the number of iterations to 100 does not help:

```
 l1 = 1;
l2 = 1;
l3 = 1.5;
x = -1.5;
y = -0.776;
z = 1.1;
FindRoot[{x == Cos[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    y == Sin[t1] (l1 + l2 Cos[t2] + l3 Cos[t2 + t3]),
    z == l2 Sin[t2] + l3 Sin[t2 + t3]}, {t1, 0.1},{t2, 0.1},{t3, 1.5},
  MaxIterations->100]

FindRoot::"cvnwt":
    "Newton's method failed to converge to the prescribed accuracy after \
100 iterations."
```

Alternatively, if you like matlab, you can use

```
>> [t1,t2,t3]=solve('cos(t1)*(1.0+cos(t2)+cos(t2+t3))=3.0','sin(t1)\
*(1.0+cos(t2)+cos(t2+t3))=0.0','sin(t2)+sin(t2+t3)=0.0')
```

```
t1 =

[ 3.1415926535897932384626433832795]
[ 3.1415926535897932384626433832795]
[                                 0]


t2 =

[ 3.1415926535897932384626433832795+1.3169578969248167086250463473080*i]
[ 3.1415926535897932384626433832795-1.3169578969248167086250463473080*i]
[                                                                      0]


t3 =

[ -2.6339157938496334172500926946159*i]
[  2.6339157938496334172500926946159*i]
[                                    0]
```

Clearly, the third solution is the right one since imaginary joint angles aren't allowed.

5. I used the mathematica code called "pieper.nb" available in the "Handouts" section of the course web page.